

VŠB – Technická univerzita Ostrava

Fakulta elektrotechniky a informatiky

Katedra informatiky

Generátor libovolných průběhů napájecích napětí

Power line arbitrary waveform generator

Zadání bakalářské práce

Student: **Martin Minář**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: **Generátor libovolných průběhů napájecích napětí**
Power Line Arbitrary Waveform Generator

Zásady pro vypracování:

Cílem práce je využití programovatelného zdroje PCR 1000 LA k realizaci generátoru libovolných průběhů napájecích napětí.

1. Seznamte se s programovatelným zdrojem PCR 1000 LA.
2. Nastudujte problematiku ovládání zdroje přes rozhraní RS-232.
3. Zvolte vhodné programové prostředí, implementujte uživatelské rozhraní umožňující ovládání zdroje ze vzdáleného počítače a generování zadaných průběhů napětí.
4. Simulujte případ napájení PC ze záložního zdroje a porovnejte jej s harmonickým napájením.
5. Zhodnoťte dosažené výsledky.

Seznam doporučené odborné literatury:

- [1] Manuál zdroje PCR 1000 LA, http://www.kikusui.co.jp/kiku_manuals/P/PCR_LA_EB.PDF.
- [2] Kainka, B.: Elektronika s podporou PC, HEL 2004.
- [3] Lukeš, P.: Poklesy napětí v síti nn a spolehlivost výpočetní a řídicí techniky, AUTOMA 1/2008.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

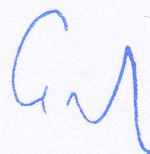
Vedoucí bakalářské práce: **Ing. Petr Orság, Ph.D.**

Datum zadání: 01.09.2013

Datum odevzdání: 07.05.2014



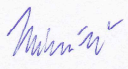
doc. Dr. Ing. Eduard Sojka
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

„Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.“

V Ostravě dne: 30.4.2014

Podpis: 

Zde bych chtěl poděkovat panu Ing. Petru Orságovi, Ph.D. za pomoc při tvorbě bakalářské práce.

Abstrakt

Tato bakalářská práce je zaměřena na generování libovolných průběhů napájecích napětí a simulaci poruchových stavů zdroje Kikusui PCR 1000 LA. V práci jsem se věnoval jak teoretickému popisu, tak praktickým zapojením a programování napájecích průběhů. V závěru jsem zhodnotil dosažené výsledky.

This work is focused on generating arbitrary waveform supply voltages and simulation of fault conditions Kikusui PCR 1000 LA. In my work I paid attention to theoretical description, practical wiring and programming of the supply waveforms. In the end, I reviewed the results.

Klíčová slova

Generátor libovolných průběhů napájecích napětí, Kikusui PCR 1000 LA, grafické uživatelské rozhraní, virtuální instrumentace, sériové rozhraní, GPIB, ethernet, napětí, sinus, obdélníkový signál, trojúhelníkový signál, přepětí, podpětí, simulace, zdrojový kód.

Power line arbitrary waveform generator, Kikusui PCR 1000 LA, graphical user interface, virtual instrumentation, serial interface, GPIB, ethernet, voltage, sine, rectangular signal, triangular signal, overvoltage, undervoltage, simulation, source code.

Seznam symbolů a zkratek

A/D převodník Analogově-digitální převodník

D-Sub Konektor pro přenos signálů

EISA Rozšíření ISA

Ethernet Soubor standardů pro lokální počítačové sítě

GND Uzemnění

GPIO General Purpose Interface Bus

GUI Graphical user interface (grafické uživatelské rozhraní)

I (A) Proud

IEEE Institute of Electrical and Electronics Engineers (Institut pro elektrotechnické a elektronické inženýrství)

ISA Typ počítačové sběrnice

ISO/OSI Referenční model komunikace v PC sítích pomocí vrstevnatého modelu

I/O Vstupně-výstupní

k Řád harmonické

LSB Nejméně významný bit

MSB Nejvýznamnější bit

P (W) Výkon

PC Personal computer (osobní počítač)

PCI Počítačová sběrnice pro připojení různých periférií

PLC Systém pro přenos dat

PXI Systém pro měření

RS-232 Standard sériového portu

RxD	Lína pro přenos dat u sériového portu
S (VA)	Zdánlivý výkon
SCADA	Supervisory kontrol and data acquisition (dispečerské řízení a sběr dat)
t (s)	Čas
TCP/IP	Rodina protokolů
THD_I	Činitel harmonického zkreslení proudu
THD_U	Činitel harmonického zkreslení napětí
TxD	Lína pro přenos dat u sériového portu
U (V)	Napětí
UART	Universal Serial Receiver Transmitter
U_m (V)	Amplituda střídavého napětí
UPS	Uninterruptible Power Source (nepřerušitelný zdroj energie)
USB	Universal serial bus
U_{mk}	Amplituda k-té harmonické
VXI	Druh sériového rozhraní
W (J)	Práce
ψ_0	Počáteční fáze střídavého napětí
ψ_k	Počáteční fáze k-té harmonické
ω	Úhlová frekvence

Obsah

1	Úvod.....	1
2	Virtuální instrumentace – základní principy	2
2.1	Klasické přístroje.....	2
2.2	Přístroje vybavené komunikačním rozhraním.....	2
2.2.1	Sériové rozhraní RS-232	3
2.2.2	Paralelní rozhraní GPIB (General Purpose Interface Bus).....	3
2.2.3	Rozhraní Ethernet.....	4
2.3	Virtuální instrumentace ve vývoji techniky.....	4
2.4	Fáze procesu instrumentace.....	5
2.4.1	Fáze sběru nebo vysílání či řízení dat	5
2.4.2	Fáze analýzy a zpracování dat.....	5
2.4.3	Fáze prezentace naměřených a analyzovaných dat	6
3	Proč sériové rozhraní?.....	7
3.1	Popis sériového rozhraní RS-232	7
3.2	Popis přenosu přes RS-232.....	8
3.3	Asynchronní komunikace	8
3.4	Sériový přenos dat	8
4	Popis zdroje PCR 1000 LA.....	10
4.1	Technické parametry	10
4.2	Princip bank.....	12
5	Požadované průběhy napájecích napětí	14
5.1	Harmonický průběh	14

5.1.1	Fázově řízený harmonický průběh	15
5.2	Pilový průběh.....	16
5.3	Trojúhelníkový průběh	16
5.4	Obdélníkový průběh	17
5.5	Obecný neharmonický průběh.....	18
5.6	Fourierův rozvoj	18
6	Poruchové stavy	20
6.1	Přepětí.....	20
6.2	Podpětí.....	20
6.3	Simulace poruchových stavů.....	20
6.4	Jak se bránit proti výpadkům v el. síti	21
6.5	Simulace napájení PC ze záložního zdroje.....	22
6.5.1	Použité přístroje a software.....	22
6.5.2	Postup měření a naměřené hodnoty	22
6.5.3	Zhodnocení a závěr měření	25
7	Uživatelské rozhraní	26
7.1	Charakteristika LabWindows/CVI	26
7.2	Charakteristika MS Visual Studia 2010	26
7.3	Srovnání LabWindows/CVI a MS Visual Studio.....	27
7.4	Práce v programu MS Visual Studio	28
7.5	Návrh uživatelského rozhraní.....	29
7.6	Popis grafického uživatelského rozhraní.....	30
7.7	Popis funkčnosti aplikace	32

7.8	Popis zdrojového kódu	32
7.8.1	Třída „MainFrom.cs“	33
7.8.2	Třída „PortCommunication.cs“	33
8	Testování aplikace	34
8.1	Testování vstupů a výstupů	34
8.2	Testování simulací poruchových stavů.....	35
8.3	Nahrávání průběhů napájecích napětí do bank zdroje	36
9	Závěr	37
9.1	Zhodnocení mé práce.....	37
9.2	Dosažené výsledky	37
9.3	Shrnutí	38
	Literatura.....	39
	Seznam příloh	40

1 Úvod

Náplní bakalářské práce je návrh grafického uživatelského rozhraní ke zdroji Kikusui PCR 1000 LA umožňující jeho ovládání ze vzdáleného počítače a využití zdroje k realizaci generátoru libovolných průběhů napájecích napětí. Zdroj je programovatelný, takže umožňuje generovat různé průběhy napájecích napětí a lze ho využít pro simulace poruchových stavů.

V práci se budu věnovat především typickým průběhům napájecích napětí a jejich základním modifikacím. Dále se budu zabývat základními poruchovými stavy, jako je podpětí nebo přepětí a jejich simulaci.

Výstupem bude aplikace s grafickým uživatelským rozhraním, umožňující ovládání zdroje z počítače, simulaci poruchových stavů a zasílání libovolných průběhů do zdroje. Na konec zhodnotím dosažené výsledky a přínos mé práce.

2 Virtuální instrumentace – základní principy

Dělení:

- Klasické přístroje
- Přístroje vybavené komunikačním rozhraním

2.1 Klasické přístroje

U klasických analogových přístrojů bez komunikačního rozhraní jako jsou voltmetry, ampérmetry, osciloskopy a další přístroje, je funkce měřicího přístroje definována jednoznačně jeho výrobcem. Koncový uživatel tedy může využít jen firmware, který je implementován v přístroji již z výroby [3].

2.2 Přístroje vybavené komunikačním rozhraním

Typicky u měřicích přístrojů vybavených komunikačním rozhraním můžeme naměřené hodnoty přebírat i bez lidského činitele, a to přímou komunikací počítače a měřicího přístroje prostřednictvím například sériové linky. Rozšiřují možnosti přístroje zejména s ohledem na zvýšení komfortu jeho ovládání a automatizaci jeho činnosti.

Proces měření se vyznačuje dvěma etapami, v první etapě se konfiguruje přístroj, nastaví se všechny parametry na hodnoty potřebné pro dané měření. Informace v první etapě jdou převážně od uživatele k přístroji, ten nastavuje hodnoty například pomocí tlačítek na čelním panelu přístroje. V druhé etapě dochází k odečítání měřených hodnot, takže informace jdou v tuto chvíli převážně od přístroje k uživateli. K tomu může sloužit displej na čelním panelu přístroje [3].

Náhrada koncového uživatele počítačem přináší dvě výhody:

- Automatizace celého měření – zefektivní proces z hlediska času
- Vyloučení lidského faktoru – eliminujeme chyby například špatně přečtenou hodnotu

Přístroje komunikují s PC nejčastěji přes:

- Sériové rozhraní RS-232
- Paralelní rozhraní GPIB
- Ethernet

2.2.1 Sériové rozhraní RS-232

Toto sériové rozhraní slouží pro sériovou komunikaci dvou zařízení např. PC s PC nebo PC s jiným přístrojem. Sériová komunikace znamená, že bity jsou vysílány za sebou, v sérii po jednom páru vodičů v každém směru. Jedná se o bezkolizní fyzickou vrstvu [4].

Značnou výhodou tohoto rozhraní je jednoduchá a levná kabeláž (stačí i jeden signálový zemní vodič pro komunikaci mezi měřicím přístrojem a PC). Tímto rozhraním je PC vybaven standardně (myš) a linka, přes kterou lze přístroj ovládat, může být poměrně dlouhá [4].

Nevýhodou je omezení na dvě zařízení, které spolu komunikují (PC a přístroj). Další nevýhody jsou malá propustnost tj. nízká přenosová rychlost, nutnost sériového řazení informace po bitech a nutnost redundantní informace pro synchronizaci přenosu – start bit, paritní bit, stop bit [4].

2.2.2 Paralelní rozhraní GPIB (General Purpose Interface Bus)

Historicky bylo zavedeno firmou Hewlett Packard v roce 1965 pod označením HP-IB. V roce 1975 bylo zdokonaleno do podoby standardu IEEE 488 a v roce 1987 do podoby ANSI/IEEE 488.1, později rozšířeno na ANSI/IEEE 488.2 standard [4].

Výhody tohoto rozhraní jsou vysoká přenosová rychlost až přes 1 MB/s a s využitím protokolu HS488 až 8 MB/s. Toto rozhraní na rozdíl od RS232 umožňuje na jeden řídicí počítač připojit více měřicích přístrojů. V případě IEEE 488.2 existence standardizovaného protokolu pro řízení přístrojů akceptovaného všemi výrobci, kteří tímto rozhraním své přístroje vybavují [3].

Nevýhody tohoto rozhraní jsou omezená vzdálenost komunikujících přístrojů (maximální délka kabelu je 20 m), nutnost vybavit počítač modulem tohoto rozhraní, dražší kabeláž [3].

2.2.3 Rozhraní Ethernet

Ethernet je v dnešní době nejrozšířenější technologie pro budování počítačových sítí typu LAN. Stal se de facto standardem pro svojí jednoduchost a vytlačil tak ostatní alternativní technologie (např. ATM). Jako konektor se používá RJ-45 pro kroucenou dvojlinku, jako přenosové médium však můžeme použít i optické vlákno nebo koaxiální kabel. Pro přístup na společné médium používá metodu CSMA/CD což je mnohonásobný přístup k médium s nasloucháním nosné a detekcí kolizí.

Výhody rozhraní jsou například připojení měřicího přístroje do lokální sítě na standardní médium (jako jsou koaxiální kabel nebo kroucená dvojlinka) v místě měření přes standardní vstup do sítě. Dále je to možnost konfigurace měřicího přístroje a zpracování naměřených dat v jiném místě lokální počítačové sítě nebo odkudkoliv z internetu, velká propustnost tohoto rozhraní (100 Mbit/s), možnost využití protokolu TCP/IP [3].

Nevýhoda je, že standardní obslužné programy jsou zatím uzpůsobeny pro využití rozhraní RS-232 nebo GPIB [3].

2.3 Virtuální instrumentace ve vývoji techniky

Podstatou virtuální instrumentace je využití PC ke komunikaci s přístrojem a doplnění o funkce, které standardně nenabízí. Typickým příkladem je hardwarové doplnění počítače o multifunkční měřicí kartu (sběrnice ISA, EISA, PCI). V oblasti softwaru je to program, který realizuje všechny funkce měřicího přístroje a plní tak vlastně úlohu firmwaru přístroje. Virtuální přístroj má GUI (grafické uživatelské rozhraní) – čelní panel virtuálního přístroje. Uživatel na něm má k dispozici ovládací prvky pro nastavení různých parametrů přístroje a indikační prvky, které informují například o výsledcích měření, jako jsou displeje nebo ukazatele hodnot.

Na rozdíl od klasického přístroje si virtuální přístroj ponechává všechny výhody počítače. Zásuvné multifunkční karty mají omezení hlavně v parametru dosažitelné vzorkovací frekvence a současnosti snímání z více kanálů. Toto omezení rychlosti je u zásuvných multifunkčních karet dáno především architekturou A/D převodníku, šířkou a časováním sběrnice mezi kartou a PC. Další možností je využití technologických procesů či technologického zařízení [3].

Propojení počítače s měřicím přístrojem umožňuje doplnit funkce definované jeho výrobcem funkcemi, které lze naprogramovat do řídicího počítače. Funkce měřicího přístroje se v tomto případě obvykle omezí na odečtení naměřených dat a jejich přenos do počítače přes komunikační rozhraní. V počítači proběhne následné zpracování naměřených hodnot a softwarově můžeme realizovat i funkce, které v měřicím přístroji nejsou implementovány. Výrazně se tak zvýší flexibilita měřicího přístroje, narazí-li uživatel na funkci, kterou přístroj nemá implementovanou, může jí pomocí softwaru na PC realizovat [3].

Výrobci měřicí techniky při analýze přišli na to, že se často z měřicího přístroje využijí jen vstupní obvody a paměť dat. Další krok vývoje techniky byl směrem k virtuálním měřicím přístrojům [3].

2.4 Fáze procesu instrumentace

Obecné schéma procesu má tři fáze:

- fáze sběru dat či řízení technologického procesu
- fáze analýzy naměřených dat
- fáze prezentace naměřených a analyzovaných dat

2.4.1 *Fáze sběru nebo vysílání či řízení dat*

Je to první fáze procesu, kde se získávají tzv. surová data, jedná se o převod měřené veličiny na elektrický signál nesoucí informaci o měřené veličině a následně jeho převod na číselnou hodnotu. Pak je tato informace předána počítači a buď uložena v paměti, nebo zobrazena na monitoru PC.

Fáze sběru dat či řízení technologického procesu v podstatě používá tyto způsoby sběru dat přes zásuvné multifunkční karty, rozhraní RS-232 nebo jiné sériové rozhraní, rozhraní GPIB, VXI, PXI měřicí systémy, PLC, průmyslové I/O systémy, systémy snímání obrazu, SCADA systémy (dispečerské řízení a sběr dat).

2.4.2 *Fáze analýzy a zpracování dat*

Druhá fáze procesu, ve které dochází k odstranění rušení, statistickému vyhodnocení (výpočet střední nebo efektivní hodnoty), výpočtu nepřímě měřených veličin a indikaci stavu zařízení.

Fáze analýzy naměřených dat se skládá z digitálního zpracování signálu například digitální filtrace nebo výpočet frekvenční analýzy, statistiky, operace s datovými poli.

2.4.3 Fáze prezentace naměřených a analyzovaných dat

Naměřené a analyzované hodnoty prezentujeme ve formě protokolů z měření, grafů atd.

- grafické rozhraní k uživateli
- síťové aplikace
- archivace v souborech
- tisk

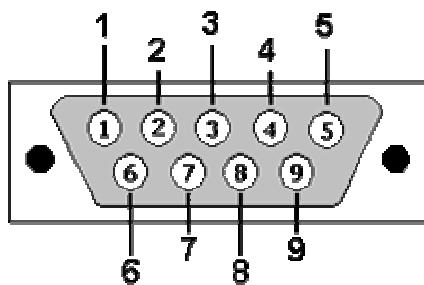
3 Proč sériové rozhraní?

3.1 Popis sériového rozhraní RS-232

V první řadě proč používat port RS-232, když už máme USB? Na tuto otázku je jednoduchá odpověď. Důvodem použití RS-232 v době, kdy už máme USB je, že námi používaný zdroj PCR 1000 LA toto rozhraní v době pořízení jako jediné podporoval. Existují však redukce RS-232 na USB pro počítače, které tento port nemají.

Poslední standard z roku 1969 RS-232C se používá jako komunikační rozhraní osobních počítačů a dalších elektronických přístrojů. Komunikace probíhá po sériové lince, jednotlivé bity přenášených dat jsou vysílány postupně za sebou po jednom páru vodičů v každém směru. Je provozováno na fyzické vrstvě modelu ISO/OSI. V současné době je rozhraní RS-232 nahrazováno sériovým rozhraním USB (Universal Serial Bus). RS-232 definuje jak přenést mezi zařízeními určitou sekvenci bitů, nezabývá se vyššími vrstvami komunikace [4].

Na počítači nalezneme sériovou linku RS-232 nejčastěji jako konektor D-Sub typu DE-9 M (samec). Existují převodníky mezi USB a RS-232, které mají oproti originální lince delší dobu odezvy – v některých aplikacích můžou nastat problémy [4].



Obr. 1: Konektor RS-232

- 1.) DCD – Data Carried Detect
- 2.) RxD – Receive Data
- 3.) TxD – Transit Data
- 4.) DTR – Data Terminal Ready
- 5.) GND – System Ground

- 6.) DSR – Data Set Ready
- 7.) RTS – Request To Send
- 8.) CTS – Clear To Send
- 9.) RI – Ring Indicator

3.2 Popis přenosu přes RS-232

Asynchronní sériová komunikace je definována pro přenos dat. Pořadí přenosu bitů jde od nejméně významného LSB po nejvýznamnější bit MSB. Obvykle se používá 8 datových bitů. Logická 0 a logická 1 je reprezentována pomocí dvou úrovní napětí, nejčastěji používaná bipolární napětí jsou pro log. 1 napětí -12 V a pro log. 0 napětí +12 V. Rozhraní má základní tři vodiče pro příjem, vyslání a společnou zem GND, ty jsou doplněny ještě dalšími vodiči, které slouží k řízení přenosu – mohou a nemusí být používány (zapojeny) [4].

3.3 Asynchronní komunikace

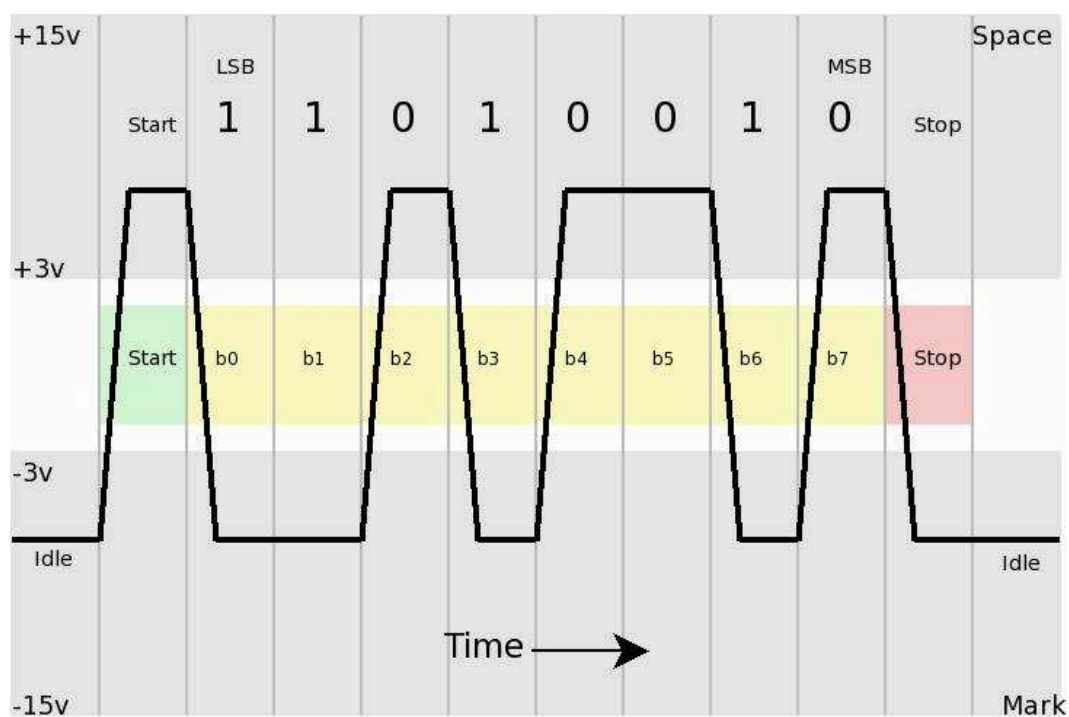
Přijímač musí začít přijímat ve správný okamžik, proto probíhá synchronizace. Vysílač pošle nějaká definovaná data po datovém vodiči, když je přijímač přijme tak se zasyynchronizuje. U RS-232 každé sekvenci bitů předchází tzv. start bit (logická hodnota na lince se přepne do opačného stavu). Po datových bitech následuje paritní bit a za ním nakonec stop bit – linka je opět v klidovém stavu [4].

3.4 Sériový přenos dat

Vlastní účel sériového portu bylo přenášet data přes modemy. Svědčí o tom i označení jako „Ring Indicator“ (indikátor zvonění) nebo „Data Terminal Ready“ (připravenost koncového zařízení). Postupem času se však sériový port více osvědčoval i v jiných oblastech, největší rozšíření byla počítačová myš na portu RS-232. Dnes můžou na portu RS-232 pracovat zařízení s vlastním mikrokontrolérem, které z portu odebírá napájecí napětí [1].

Účel sériového portu je přenos dat na linkách TxD a RxD, data jsou hardwarem portu převáděna na sériový tok bitů a vysílána na cestu. Na druhé – přijímací straně,

vytvoří druhý port ze sériového toku dat opět paralelní datové bajty a ty jsou dále zpracovány počítačem. Vyslání jednoho znaku přes sériový port vyžaduje značné množství času (milisekundy). Sériový port v počítači obsahuje součástku univerzální sériový vysílač/přijímač (Universal Serial Receiver Transmitter, UART), sem se předávají zprávy a samostatně se vysílají, naopak UART samostatně přijímá zprávy a ukládá je tak, aby bylo možné je jednou akcí vyzvednout. Typický je pro sériový port přenos dat z jednoho počítače do druhého. Posíláme většinou textová data, ale i obrázky nebo jen jednotlivé bajty, příklad viz obrázek 2. Blíže o sériovém přenosu dat viz literatura [4].



Obr. 2: Přenos znaku „K“ přes RS 232 bez parity a s jedním stopbitem

4 Popis zdroje PCR 1000 LA

4.1 Technické parametry

Série PCR-LA je rozšíření série PCR-L napájecích zdrojů, díky kombinaci vysokorychlostního lineárního zesilovače a syntetizéru libovolných průběhů může PCR-LA série dosáhnout „prakticky“ harmonického průběhu napětí. Je vybaven funkcemi pro simulaci poruchových stavů a měřicími funkcemi, které lze využít pro testování elektrických zařízení. Označení PCR 1000 LA znamená 1kVA tj. 10 A pro 100 V na výstupu, 5 A pro 200 V na výstupu [2]. Parametry zdroje jsou uvedeny v následující tabulce.

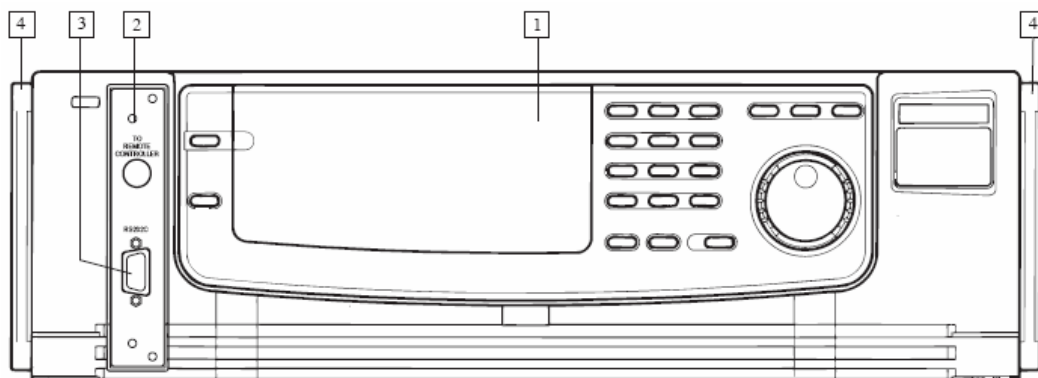
Tab. 1: Základní parametry zdroje

Mód výstupního napětí	Rozsah	Minimální hodnota	Maximální hodnota	Proudová zatížitelnost
AC nebo AC-S	200 V	1.00 [Hz]	999.9 [Hz]	10 A
	100 V			5 A
DC	200 V	Nastavení není možné		10 A
	100 V			5 A
AC + DC	200 V	Nastavení v tomto módu není povoleno		10 A
	100 V			5 A

Vlastnosti:

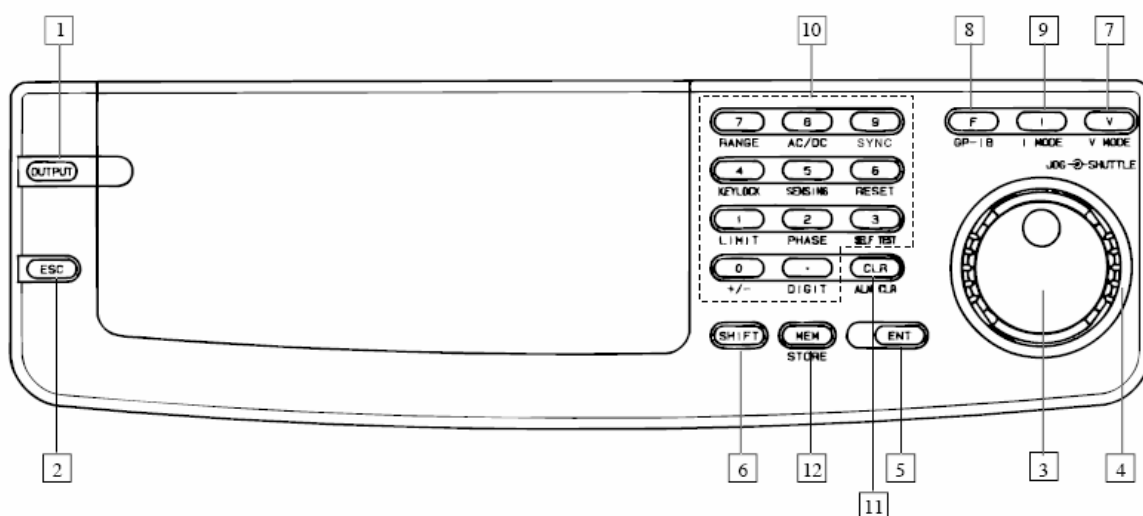
- Simulace různých druhů napájení
- Generování sekvencí průběhů
- Stejnoseměrný i střídavý výstup
- Nastavování proměnné hodnoty výstupní impedance
- Měření výstupních veličin

Zdroj je dodáván s třížilovým kabelem s PVC ochranou. Zdroj je vybaven portem RS232 pro komunikaci s PC. Pro ovládání zdroje máme k dispozici čelní panel. Pokud ovládáme zdroj přes počítač, jeho ovládání na čelním panelu je mimo provoz (musí být vypnutý výstup). Při paralelním zapojení identických modelů PCR-LA může být zapojeno až 5 jednotek. Zdroj je dále vybaven úchyty pro přenos, které jsou vysunovací [2].



Obr. 3: Přední strana zdroje Kikusui PCR 1000 LA [2]

- 1.) Kontrolní panel
- 2.) Dálkové ovládání
- 3.) RS 232 konektor
- 4.) Úchyty



Obr. 4: Čelní panel zdroje Kikusui PCR 1000 LA [2]

- 1.) OUTPUT – zapnutí/vypnutí výstupu
- 2.) ESC – escape
- 3.) JOG – pro nastavení např. napětí nebo frekvence
- 4.) SHUTTLE – pro nastavení např. napětí nebo frekvence

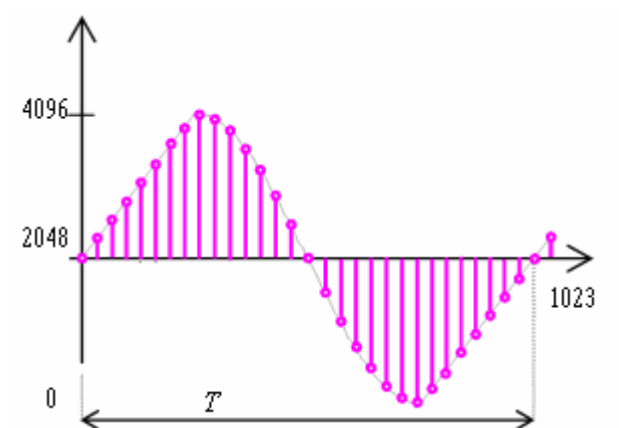
- 5.) ENT – potvrzuje klíčové operace
- 6.) SHIFT – umožní nastavit funkce s modrými nápisy
- 7.) V – vybírá mód nastavení napětí nebo napěťový rozsah
- 8.) F – vybírá mód nastavení frekvence nebo frekvenční rozsah
- 9.) I – vybírá aktuální limit módu nastavení
- 10.) Numerické klávesy
- 11.) CLR – ruší nastavené hodnoty a nastaví předchozí hodnoty
- 12.) MEM – čte nastavené napětí a frekvenci z paměti

Řada zdrojů PCR-LA obsahuje následující typy zdrojů: PCR 500 LA, PCR 1000 LA, PCR 2000 LA, PCR 4000 LA, PCR 6000 LA. Jednotlivé typy se liší hodnotami výstupu.

4.2 Princip bank

Jelikož je zdroj programovatelný, obsahuje 14 bank pro ukládání uživatelem definovaných průběhů napájecích napětí. Tyto banky jsem využil v mé aplikaci, když jsem do bank ukládal zadané průběhy. Přepínání mezi bankami pro nastavení výstupu se provede příkazem WAVEBANK X, kde X může nabývat hodnoty 1-14 pro konkrétní banku. Průběh se pak do banky ukládá příkazem WAVE X, i, průběh, kde X je číslo banky, i je pozice v bance, která nabývá hodnot 0 - 1023 a průběh je hodnota průběhu pro konkrétní pozici.

Dalšími příkazy pro práci s bankami jsou například WAVE X, X?, jenž vrátí data uložená v zadané bance a adrese a příkaz WAVEBANK?, který vrací číslo aktuálně nastavené banky. V případě překročení maximální hodnoty, která může být v bance nastavena, začne zdroj pískat chybový kód.



Obr. 5: Struktura banky

5 Požadované průběhy napájecích napětí

Mezi nejčastěji používané průběhy střídavého napájecího napětí patří:

- Harmonický průběh (sinusový průběh)
- Pilový resp. trojúhelníkový průběh
- Obdélníkový průběh
- Obecný neharmonický průběh (složený sinusový průběh)

5.1 Harmonický průběh

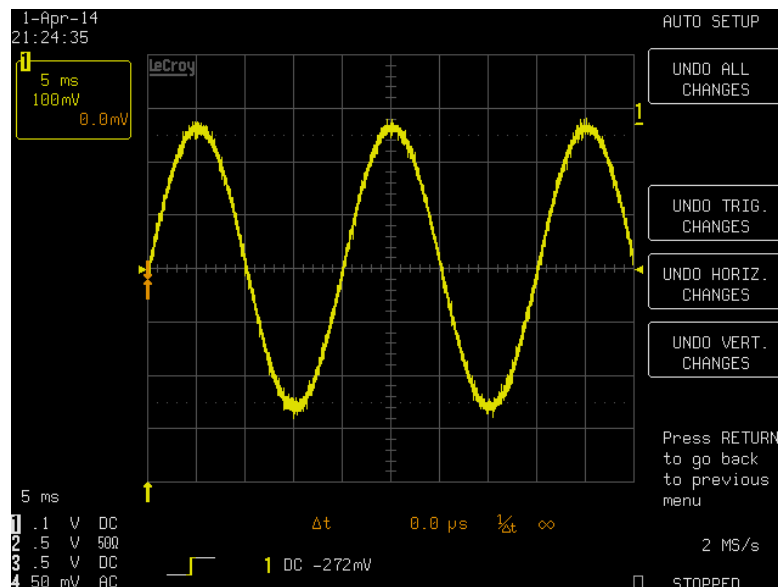
Harmonický průběh definovaný pomocí funkce sinus je goniometrická funkce, matematicky jí můžeme definovat pomocí pravoúhlého trojúhelníku jako poměr protilehlé odvěsny a přepony. Funkci Sinus nejčastěji označujeme zkratkou sin (inverzní funkci pak arkus sinus). Grafem funkce sinus je tzv. sinusoida.

Harmonický průběh střídavého napětí je definován takto:

$$u(t) = U_m \cdot \sin(\omega t + \psi_0),$$

U_m je amplituda střídavého napětí, ω je úhlová frekvence (kmitočet), ψ_0 je počáteční fáze střídavého napětí.

Střídavé napětí nebo střídavý proud nemusejí mít vždy sinusový průběh, průběh může být i např. pilový nebo obdélníkový. Střídavé napětí zobrazené na osciloskopu může mít různé tvary.

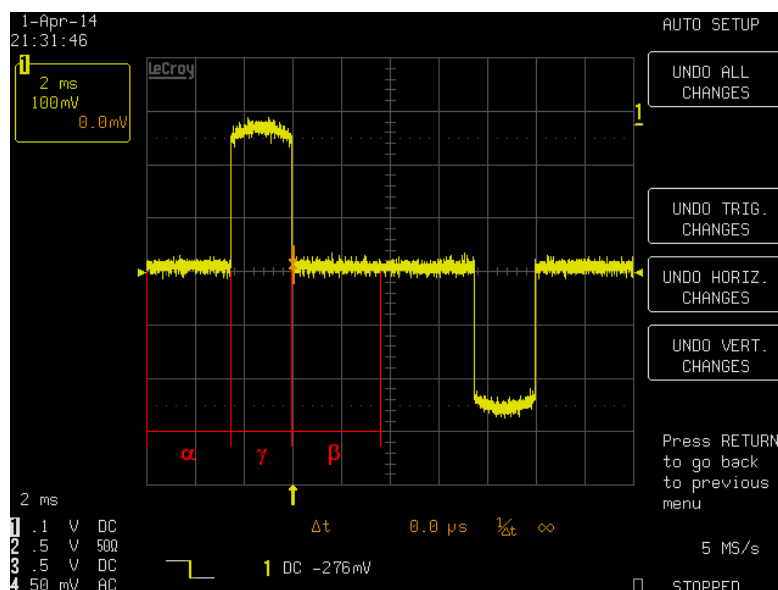


Obr. 6: Sinusový signál s nulovou počáteční fází

Příklad harmonického průběhu je na obrázku 6 (nastavené $U = 30$ V, převod izolačního zesilovače je 160). Sinusový průběh má i střídavý proud.

5.1.1 Fázově řízený harmonický průběh

Je variantou harmonického sinusového průběhu, kdy ořízneme průběh po časové ose tzv. souměrné fázové řízení (α je $0^\circ - 180^\circ$). Dostaneme pak například průběh jako na následujícím obrázku. Pro nastavené α a $\beta = 0$ je průběh harmonický.



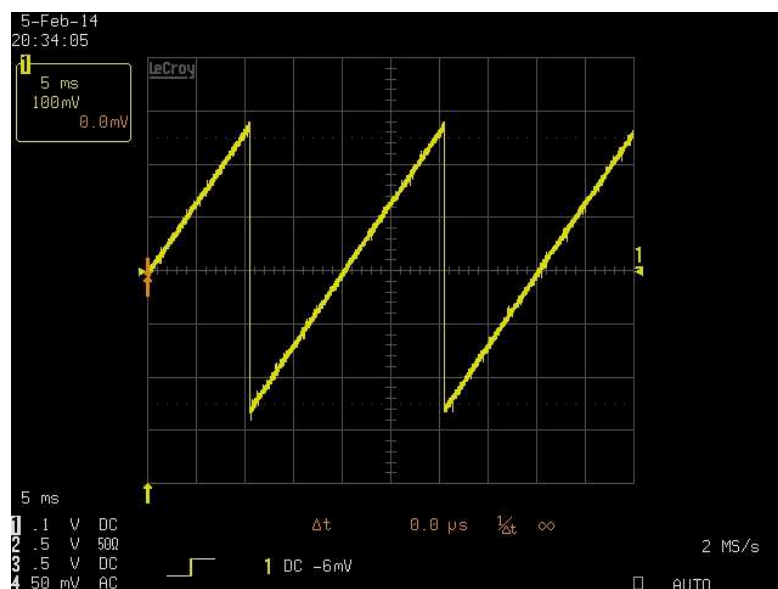
Obr. 7: Fázově řízený sinusový průběh: $\alpha + \beta + \gamma = T/2$

5.2 Pilový průběh

Pilový signál se vyznačuje plně zastoupeným počtem vyšších harmonických. V praxi ho naprogramujeme pomocí cyklu for. Tvarem se podobá opravdové pile (jedna hrana jde šikmo nahoru a druhá kolmě dolu) odtud název, narozdíl od signálu trojúhelníkového, který má tvar rovnoramenného trojúhelníku. Skokový doběh spolu s ostrým zlomem obsahuje nekonečné množství kmitočtů, proto je amplitudové spektrum také nekonečné. Fourierův rozvoj pilového průběhu je dán vzorcem:

$$u(t) = \frac{2}{\pi} \sum_{k=1}^{\infty} \left((-1)^k \frac{\sin(2\pi k f t)}{k} \right)$$

Pilový průběh je dán na intervalu $\langle 0; T \rangle$ rovnicí přímky $p: Tx - Ty = 0$.

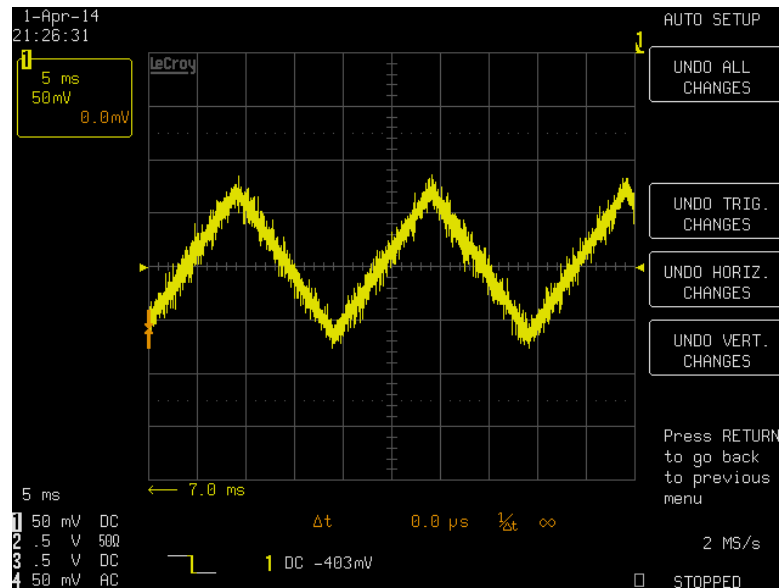


Obr. 8: Pilový signál

5.3 Trojúhelníkový průběh

Spektrum trojúhelníkového signálu neobsahuje skoky, obsahuje výrazně menší amplitudy vyšších harmonických a je nekonečné. Má tvar rovnoramenného (rovnostanného) trojúhelníku. Signál samotný se skládá z lineárního náběhu a doběhu, jež se opakují. Fourierův rozvoj trojúhelníkového průběhu je dán vzorcem:

$$u(t) = \frac{8U_m}{\pi^2} \left(\sin(\omega t) - \frac{1}{3^2} \sin(3\omega t) + \frac{1}{5^2} \sin(5\omega t) - \dots \right), k = 1, 3, 5, 7, 9, \dots$$



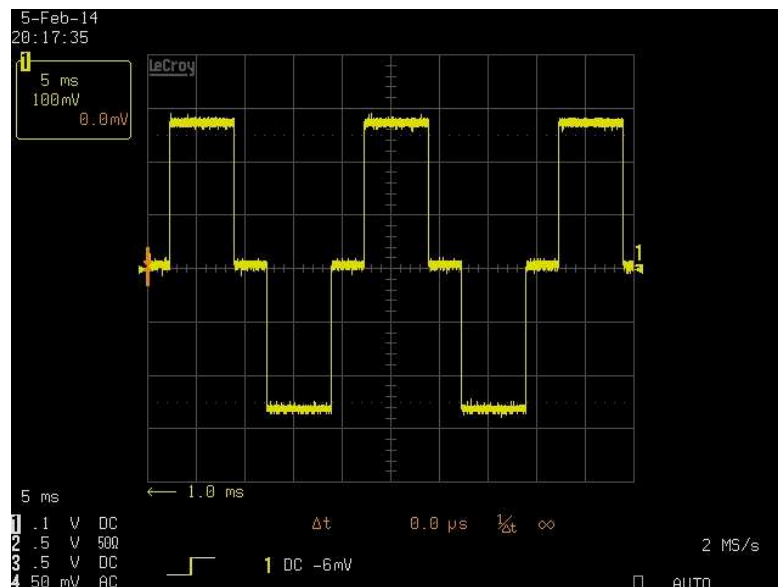
Obr. 9: Trojúhelníkový signál

Trojúhelníkový průběh je dán na intervalu $\langle 0; T/4 \rangle$ rovnicí přímky $p: Tx - Ty = 0$, na intervalu $\langle T/4; T/2 \rangle$ pak rovnicí $p: Tx + Ty - \frac{T^2}{2} = 0$, na intervalu $\langle T/2; 3T/4 \rangle$ je dán rovnicí $p: Tx + Ty - \frac{T^2}{2} = 0$ a na intervalu $\langle 3T/4; T \rangle$ je dán rovnicí přímky $p: Tx - Ty - T^2 = 0$.

5.4 Obdélníkový průběh

Obdélníkový průběh tvoří dvě konstantní úrovně nulové a kladné polarity a jeho amplitudové spektrum obsahuje nekonečně mnoho složek. Sčítáním sinusových vln donekonečna, budeme získávat stále hranatější průběh, který nakonec získá obdélníkový tvar. Spektrum obdélníkového signálu obsahuje nekonečné množství vyšších harmonických. Fourierův rozvoj obdélníkového průběhu je dán vzorcem:

$$\begin{aligned} u(t) &= \frac{4U_m}{\pi} \left(\frac{\cos(\omega t_0)}{1} \sin(\omega t) + \frac{\cos(3\omega t_0)}{3} \sin(3\omega t) + \frac{\cos(5\omega t_0)}{5} \sin(5\omega t) + \dots \right) \\ &= \frac{4U_m}{\pi} \sum_{k=1}^{\infty} \left(\frac{\cos(k\omega t_0)}{k} \sin(k\omega t) \right), k = 1, 3, 5, 7, 9, \dots \end{aligned}$$



Obr. 10: Obdélníkový signál

Obdélníkový průběh je dán na intervalu $\langle 0; T/4 \rangle$ rovnicí přímky $p: -\frac{T}{4}y = 0$, na intervalu $\langle T/4; T/2 \rangle$ pak rovnicí $p: -Ty + \frac{T^2}{2} = 0$, na intervalu $\langle T/2; 3T/4 \rangle$ je dán rovnicí $p: -\frac{T}{4}y = 0$ a na intervalu $\langle 3T/4; T \rangle$ je dán rovnicí přímky $p: -Ty - \frac{T^2}{2} = 0$.

5.5 Obecný neharmonický průběh

Je součet více sinusových signálů, jejich součet můžeme vyjádřit Fourierovou řadou viz kapitola 5.6. Složený sinusový průběh je aproximace neharmonických průběhů, harmonickými průběhy.

5.6 Fourierův rozvoj

Fourierův rozvoj vyjádřený v amplitudovém tvaru je tvořen nulovou složkou a harmonickými s parametry: k řád harmonické, U_{mk} amplituda k -té harmonické, ψ_k počáteční fáze k -té harmonické a $\omega = 2\pi f$ je úhlová rychlost základní harmonické. Periodě T veličiny u odpovídá kmitočet $f = 1/T$. Po dosazení Fourierova rozvoje veličiny do definice efektivní hodnoty a využití ortogonalit goniometrických funkcí a integraci na periodě T získáme Parsevalův vztah pro výpočet efektivní hodnoty

$$U = \sqrt{\sum_{k=0}^{\infty} U_k^2}$$

kde $U_k = \frac{U_{mk}}{\sqrt{2}}$ je efektivní hodnota harmonické složky.

Fourierovy rozvoje mají teoreticky nekonečný počet členů, ale z praktického hlediska se stačí omezit na několik nejvýznamnějších harmonických, čímž získáme aproximovaný průběh.

6 Poruchové stavy

6.1 Přepětí

Přepětí napětí je napětí, které je vyšší než amplituda jmenovitého napětí, může mít různou velikost a časový průběh, liší se rovněž příčinou, kterou vznikne. Můžeme je dělit podle původu vzniku na **provozní** (vnitřní) přepětí a **atmosférická** (vnější) přepětí [11].

Dělení přepětí podle časového průběhu:

- Dočasné
- Přechodné
- Trvalé

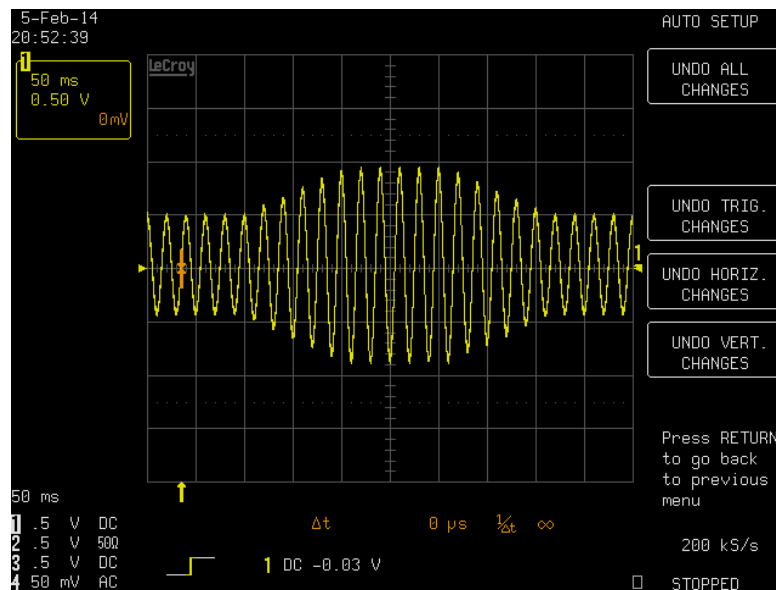
Provozní přepětí nastává např. při poruchových stavech (zkrat), ztrátě zatížení (dynamické přepětí).

6.2 Podpětí

Podpětí je opak přepětí. Na rozdíl od přepětí, kde je napětí příliš vysoké, u podpětí je napětí příliš nízké, což sice nemůže poškodit elektrické spotřebiče, ale spotřebiče při podpětí zcela nefungují nebo nefungují správně [11].

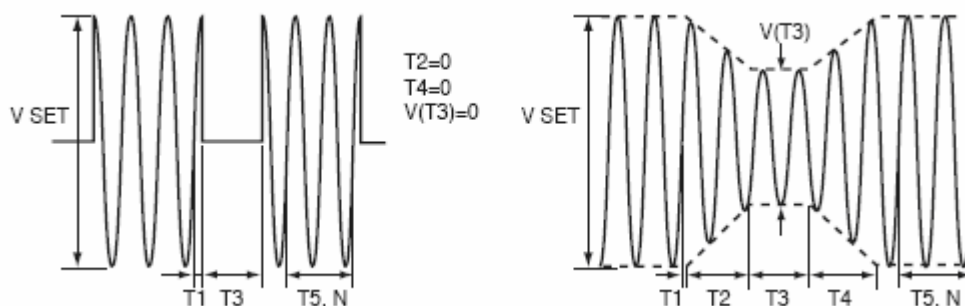
6.3 Simulace poruchových stavů

V bakalářské práci jsem měl rovněž za úkol simulovat jednotlivé poruchové stavy. Simulace jsem naprogramoval pomocí funkcí zdroje k tomu určených (viz. Manuál zdroje PCR 1000 LA). Simulace poruchových stavů jsem rovněž naprogramoval do uživatelského rozhraní.



Obr. 11: Simulace poruchového stavu - přepětí

Funkce SIMMODE nastavuje simulaci poruchového stavu, je povolena v případě, že je výstup odpojen. Po funkci SIMMODE se nastavují jednotlivé doby $T1 - T5$, počet cyklů, hodnotu napětí v čase $T3$ a nakonec normální napětí příkazem VSET (nastavené hodnoty viz. příklad testu v kapitole 8.2). Po nastavení potřebných hodnot zapneme výstup příkazem OUT 1 a následně simulaci pustíme příkazem SIMRUN. Simulaci ukončíme příkazem SIMSTOP a pak odpojíme výstup příkazem OUT 0.



Obr. 12: Nastavení parametrů simulace – přerušení a podpětí [2]

6.4 Jak se bránit proti výpadkům v el. síti

Proti výpadkům v elektrické síti se můžeme bránit použitím tzv. UPS, které v případě výpadku nahradí elektrický zdroj. UPS (z anglického Uninterruptible Power Source) v českém překladu „nepřerušitelný zdroj energie“ nám při výpadku zajišťuje

souvislou dodávku elektrické energie, používá se u zařízení, která vyžadují neustálý nepřerušovaný provoz (počítače ve firmách, nemocnicích, letištích apod.). Zapojuje se mezi primární zdroj elektrické energie a zařízení, které chceme chránit. UPS je vlastně akumulátor, který se nabíjí není-li dodávka el. energie přerušena, slouží i jako ochrana proti podpětí. V případě výpadku se pak akumulátor vybíjí.

6.5 Simulace napájení PC ze záložního zdroje

V souladu s bodem zadání 4 jsem simuloval případ harmonického napájení z UPS zdrojem Kikusui PCR 1000 LA. Výsledky měření jsou uvedeny v tabulce 2.

6.5.1 Použité přístroje a software

- UPS APC Cyberfort 500
- Zdroj Kikusui PCR 1000 LA
- Izolační zesilovač Dewetron DAQP-HV
- Klešťová sonda Chauvin Arnoux C35N
- Osciloskop LeCroy LT264

6.5.2 Postup měření a naměřené hodnoty

K napájení počítače jsem použil UPS, dále zdroj Kikusui a síťové napájení. Napětí jsem měřil přes izolační zesilovač a proud jsem měřil klešťovou sondou. Pro záznam obrázků průběhů napětí a proudu jsem použil osciloskop a data jsem vyčetl pomocí aplikace vytvořené v LABWindows/CVI, která umožňuje exportovat data do textového souboru. Vyčtená data jsem následně zpracoval v MS Excelu.

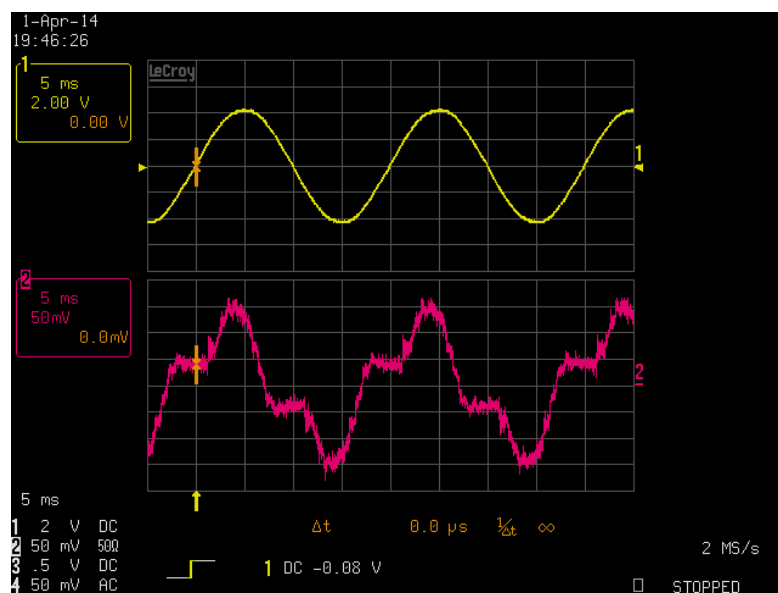
Nejdříve jsem počítač připojil do sítě, kde je 230 V a napětí má harmonický průběh. U UPS jsem využil bateriové napájení a obdélníkového průběhu napájení, který daná UPS poskytuje. Dále jsem změřil harmonické napájení ze zdroje Kikusui a nakonec obdélníkový průběh napájení ze zdroje, čímž jsem simuloval UPS napájení.

Výkon je definován jako množství práce vykonané za jednotku času: $P = \frac{W}{t}$

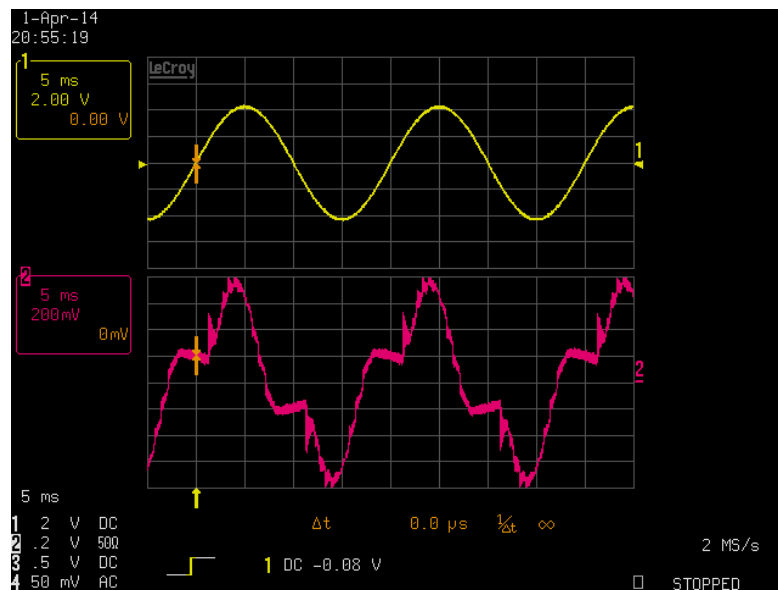
Zdánlivý výkon definujeme vztahem: $S = UI$

Tab. 2: Tabulka srovnání naměřených hodnot jednotlivých zdrojů napájení.

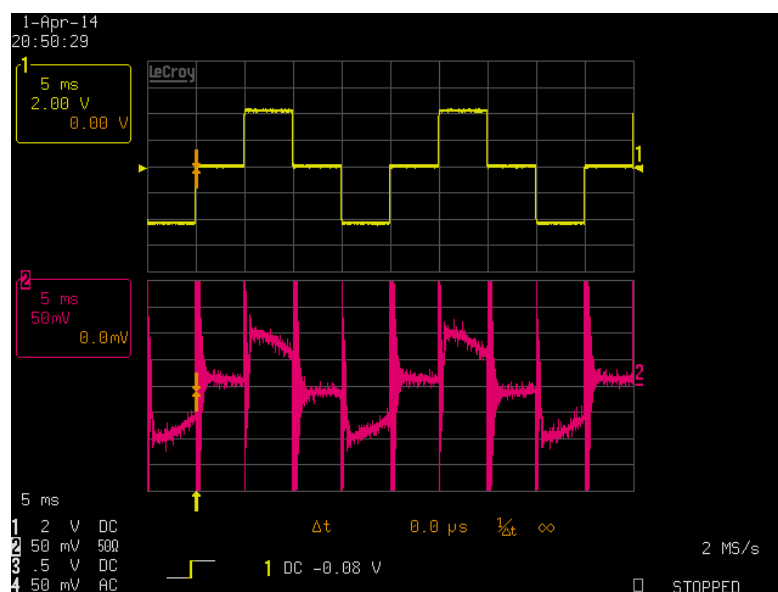
U (V)	I (A)	THD_U (%)	THD_I (%)	P (W)	S (VA)
Napájení ze sítě					
238,6	0,51	1,1	40,35	97,2	120,9
Harmonické napájení ze zdroje					
239,4	0,51	0,1	39,7	99,5	123,5
Napájení z UPS					
222,7	1,07	48,5	227,2	94,5	239,4
Napájení ze zdroje – obdélníkový průběh					
222,9	0,95	47,9	264,2	74,3	212,7



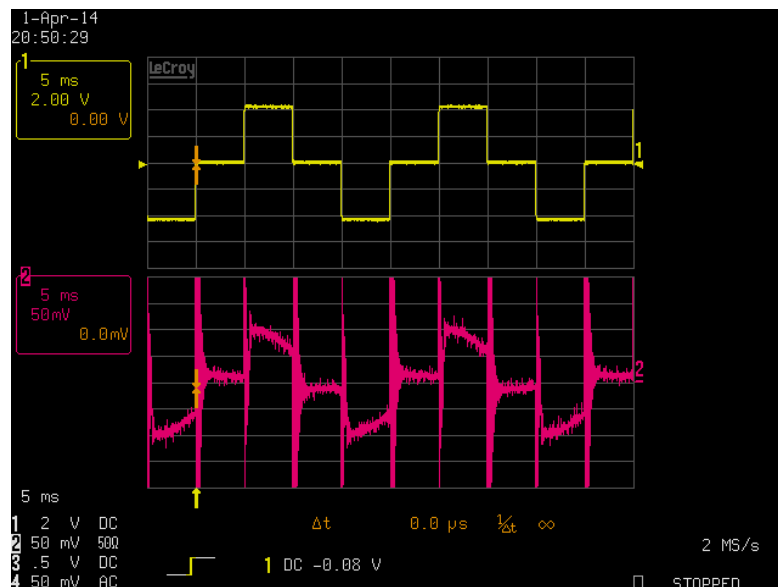
Obr. 13: Harmonický průběh napětí a průběh proudu ze síťového napájení



Obr. 14: Harmonický průběh napětí a průběh proudu ze zdroje Kikusui



Obr. 15: Obdélkový průběh napětí a průběh proudu z UPS



Obr. 16: Obdélníkový průběh napětí a průběh proudu ze zdroje Kikusui

6.5.3 Zhodnocení a závěr měření

Harmonické napájení ze zdroje jsem nastavil tak, aby odpovídalo harmonickému napájení ze sítě, u obdélníkového průběhu jsem napětí nastavil tak, aby odpovídalo napětí z UPS. Z naměřených hodnot je vidět, že činitel harmonického zkreslení napětí je u obdélníkového průběhu napětí pod 50% a u harmonického je v rozmezí 0,1 až 1,1 %. Činitel harmonického zkreslení proudu je u harmonického průběhu cca 40% v obou případech a u obdélníkového průběhu napětí je výrazně vyšší.

Dále je vidět, že zdroj Kikusui má menší harmonické zkreslení než síťové napájení. Oba výkony jak elektrický tak zdánlivý jsou u harmonického průběhu cca stejně velké. U obdélníkového průběhu jsou oba výkony nepatrně nižší u napájení ze zdroje, což je však způsobeno menším vytížením počítače. Na obrázku 15 a 16 je dobře pozorovatelný vznik vlastních kmitů, které jsou odezvou na skokovou změnu napájecího napětí.

7 Uživatelské rozhraní

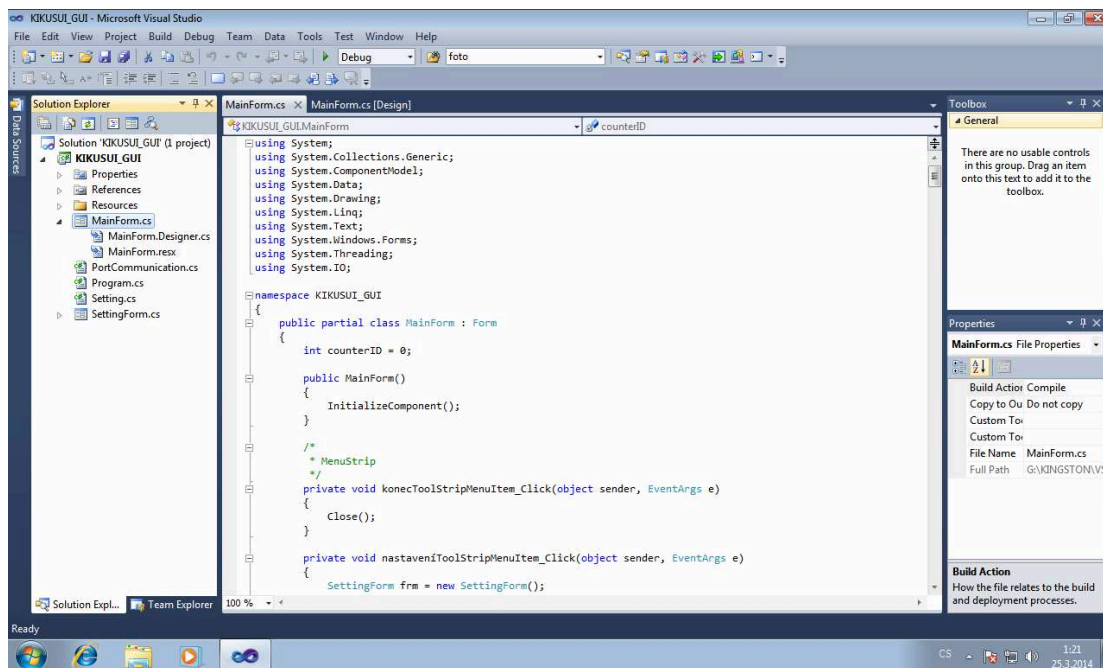
Návrh uživatelského rozhraní pro ovládání zdroje Kikusui PCR 1000 LA z počítače komunikující přes sériové rozhraní RS-232 jsem provedl dle zadání. Přes aplikaci bude možno zdroj ovládat, zapojovat a odpojovat výstup, posílat do zdroje zadané průběhy napájecích napětí a odečítat napětí a proud.

7.1 Charakteristika LabWindows/CVI

LabWindows/CVI je ANSI C programovací prostředí pro testování a měření vyvinuté firmou National Instruments. Program byl původně vydán jako LabWindows pro DOS v roce 1987. Výhodou LabWindows je, že můžeme sledovat hodnoty proměnných pomocí Breakpointů. LabWindows disponuje funkcí pro nápovědu – Utility Library. V tomto vývojovém prostředí jsem ověřoval funkčnost komunikace se zdrojem.

7.2 Charakteristika MS Visual Studio 2010

MS Visual Studio 2010 je vývojové prostředí pro vývoj aplikací zejména v C/C++ a C# od Microsoftu. Umožňuje vyvíjet jak konzolové aplikace tak aplikace s grafickým uživatelským rozhraním, obsahuje také designer pro jazyk C# k vývoji GUI aplikací. Visual Studio je sice určeno především pro jazyk C# a jemu příbuzné, umožňuje však editovat a debuggerovat jakýkoliv programovací jazyk. Podporuje např. tyto jazyky: C/C++, C#, VB.NET, Python, HTML/XHTML, JavaScript, CSS a mnoho dalších.



Obr. 17: MS Visual Studio 2010

Na obrázku je ukázka spuštěného projektu ve Visual Studiu. V horní liště se nachází klasické menu, kde najdeme věci nacházející se ve většině vývojových prostředí mimo jiné sestavení projektu, nastavení Breakpointů a jiné. Kód zkompilejeme a spustíme stisknutím klávesy F5, tím se spustí debug mód.

7.3 Srovnání LabWindows/CVI a MS Visual Studio

Důvodů proč jsem si zvolil pro implementaci MS Visual Studio je hned několik. Nejzásadnější a pro mě nejdůležitější jsou hlavně jeho velká znalost. Také je zdarma ve verzi Express, což může být výhoda pro toho, kdo nemá možnost vyšší licence, já jí ovšem mám, takže tento důvod se mě až tak netýká.

Na druhou stranu obě dvě prostředí jsou velice uživatelsky přívětivé a dobře se s nimi pracuje, LabWindows/CVI má k dispozici rozsáhlou databázi funkcí pro práci se zařízeními a nápovědy k nim. MS Visual studio má však větší uživatelskou základnu a tudíž je i více věcí popsáno v různých knihách nebo na internetu. V LabWindows/CVI jsem pracoval především proto, že jsem si chtěl vyzkoušet i tuto alternativu.

7.4 Práce v programu MS Visual Studio

Jako programové prostředí pro implementaci jsem si zvolil MS Visual Studio 2010, z důvodu jeho velké znalosti a propracovanosti tohoto prostředí. Jako programovací jazyk jsem si vybral C#, který rovněž dobře znám a ovládám. Výhodou jazyku C# je, že nám nabízí pro komunikaci přes sériovou linku třídu `System.IO.Ports`, kterou jsem použil při implementaci třídy sloužící pro komunikaci se zdrojem.

Po seznámení se všemi funkcemi zdroje jsem začal zkoušet psát nejprve základní testovací aplikace.

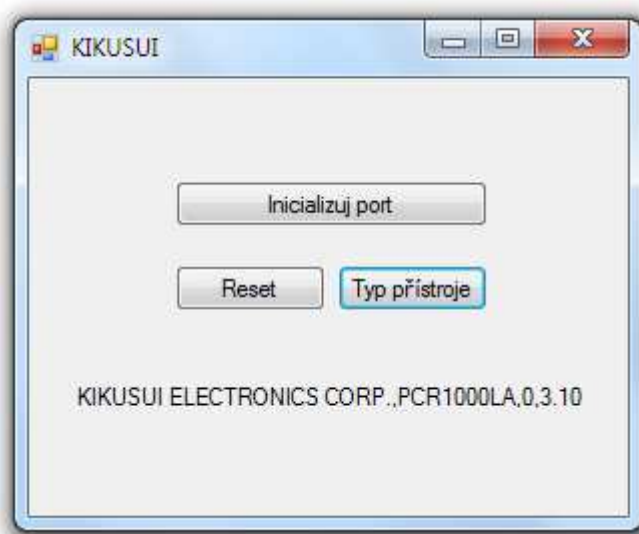
```
public static bool Init(string portN, int baudRate)
{
    portName = portN;
    try
    {
        if (port.IsOpen) return true;
        port.BaudRate = baudRate;
        port.PortName = portName;

        port.DataBits = 8;
        port.StopBits = StopBits.One;
        port.Parity = Parity.None;

        port.WriteTimeout = 5000;
        port.ReadTimeout = 5000;
        port.RtsEnable = true;
        port.DtrEnable = true;

        port.Open();
        Inited = true;
        return true;
    }
    catch (Exception ee)
    {
        port.Close();
        Inited = false;
        return false;
        // throw new Exception("Chyba pri inicializaci" + ee);
    }
}
```

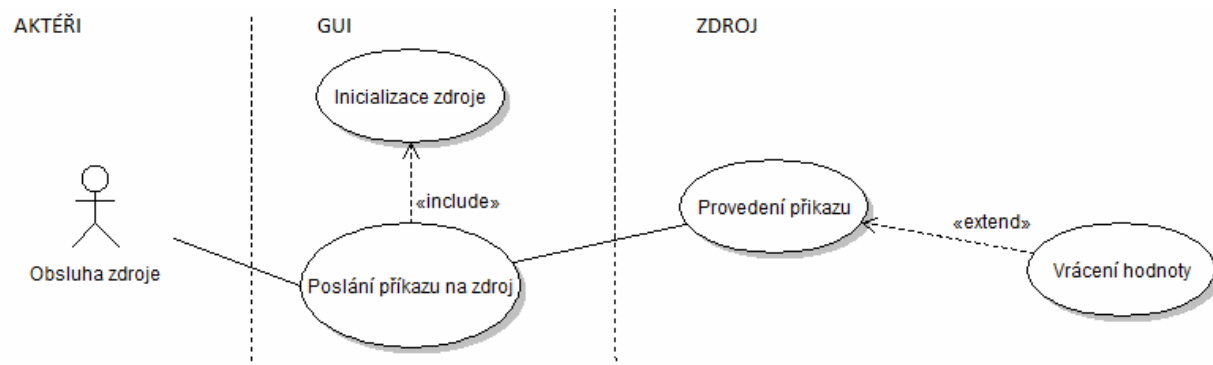
Obr. 18: Ukázka kódů, který inicializuje COM port



Obr. 19: Ukázka výstupu aplikace s využitím Windows Forms, která posílá zdroji příkaz
*idn? pro zjištění typu přístroje

Princip ovládání přístroje přes RS-232 zobrazuje následující diagram:

Komunikace probíhá podle klasické architektury klient-server. Výhodou je rozdělení jednotlivých úkolů a zodpovědnosti.



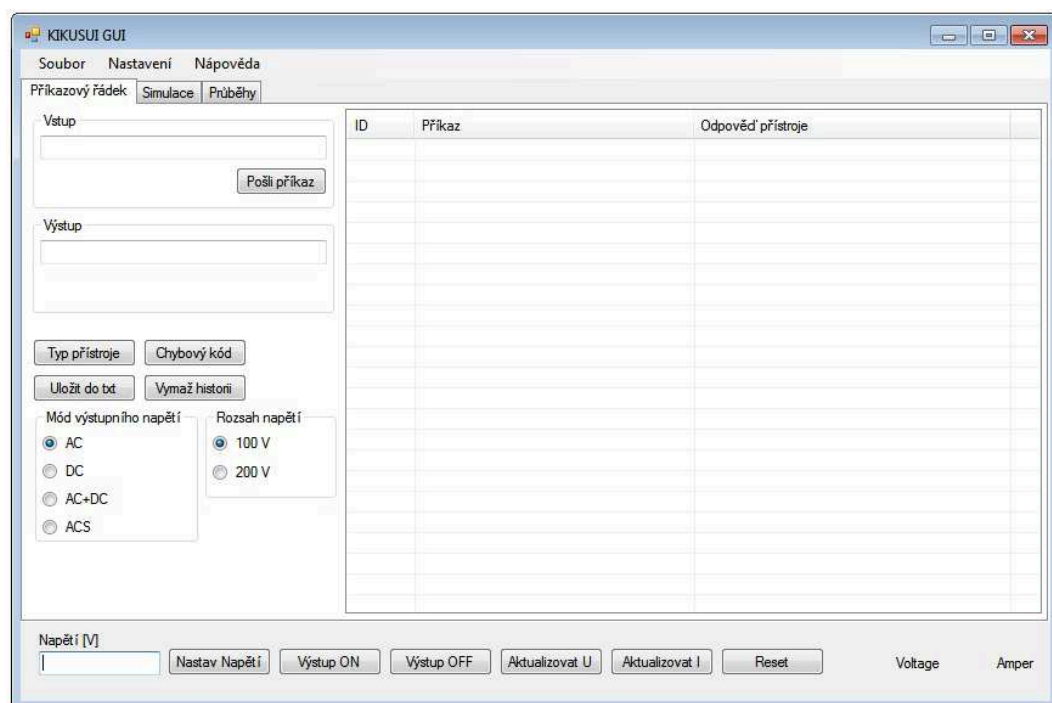
Obr. 20: Use case diagram obsluhy přístroje

7.5 Návrh uživatelského rozhraní

Samotné uživatelské rozhraní jsem se snažil navrhnout tak, aby se s ním co nejlépe pracovalo a bylo pokud možno intuitivní. Finální verze návrhu se tudíž liší od předchozích beta verzí, s kterými jsem pracoval a pořídil jsem z nich i několik náhledů. Grafické uživatelské rozhraní umožňuje ovládat zdroj Kikusui PCR 1000 LA pomocí

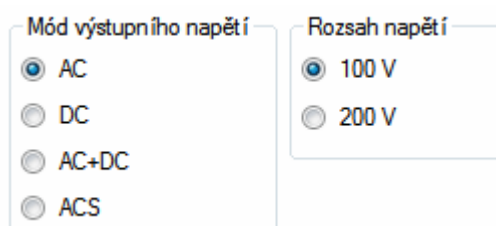
Jak jsem již psal, grafické uživatelské rozhraní jsem navrhoval v Microsoft Visual Studiu (použil jsem ověřenou verzi 2010 Professional, se kterou jsem pracoval po celou dobu bakalářského studia). Při návrhu rozhraní jsem se řídil základními principy tvorby grafických uživatelských rozhraní a to, aby bylo přehledné, snadno se v něm orientovalo na první pohled a použité písmo a barva pozadí, aby byla dobře čitelná. Použité písmo a barvu pozadí jsem nechal přednastavenou, jak dnes používá většina aplikací, vytvořených pro operační systém MS Windows. Po dokončení prvotního návrhu jsem začal s vlastním programováním aplikace.

Okno aplikace je rozděleno na tři záložky a hlavní nabídku, umístěnou v horní části okna. Horní nabídka je klasická kontextová nabídka, která se skládá z položek: Soubor, Nastavení a Nápověda. Nejdůležitější z nich je položka „Nastavení“, pod kterou se ukrývá samotné nastavení COM portu, popřípadě Baud rate. Na obrázku níže je hlavní okno aplikace, zbytek viz příloha.



-30-

Záložky aplikace jsou rozděleny na Příkazový řádek, Simulace, Průběhy. V první záložce se nachází klasický příkazový řádek, sloužící pro zaslání libovolného příkazu na náš zdroj. Dále je zde historie příkazů a možnost uložení historie příkazů zaslaných na zdroj a vrácené odpovědi zdroje. Pod příkazovým řádkem se nachází čtveřice tlačítek (button) sloužící pro získání typu přístroje, vrácení chybového kódu, uložení historie příkazů do TXT souboru a tlačítko pro vymazání historie. Nakonec je zde docela často používaná volba pro změnu módu výstupního napětí ze střídavého na stejnosměrné a také změna rozsahu výstupního napětí ze 100 V na 200 V, pro tyto funkce jsem použil přepínače.



Obr. 22: Přepínače (Radio buttons) – pokud přepneme pozici přepínače nastavení zdroje se okamžitě změní, pro funkčnost je samozřejmostí inicializace portu.

Na záložce „Simulace“ je možnost simulovat přepětí a podpětí. Po zadání příslušných hodnot T1 až T5, nastavení počtu opakování a normálního a T3 napětí, můžete simulaci spustit tlačítkem Start. Simulace se zastaví automaticky po dokončení, nebo jí můžete vypnout tlačítkem Stop, pokud simulaci necháte doběhnout až do konce, nezapomeňte vypnout výstup, buď tlačítkem „Výstup OFF“ nebo tlačítkem „Stop“.

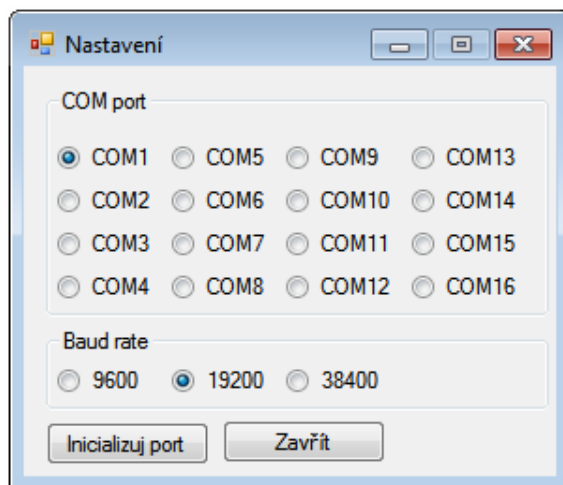
Poslední záložkou jsou samotné průběhy a jejich posílání do zdroje. Na výběr zde máme banku, do které chceme vybraný průběh nahrát. Vybraný průběh napětí můžeme různě modifikovat, například ho můžeme posunout po ose y přičtením stejnosměrné složky nebo mu změnit amplitudu. Ve spodní části nastavujeme banku pro výstup, tj. z jaké banky se na osciloskopu zobrazí průběh.

Ve spodní části okna je panel s nejpoužívanějšími funkcemi, jako jsou nastavování napětí, zapnutí a vypnutí výstupu, aktualizace hodnot a reset přístroje. Tento panel se při přepínání mezi záložkami nemění.

7.7 Popis funkčnosti aplikace

Aplikace je určena pro Windows 7, nižší verze operačního systému Windows musí mít nainstalovaný Dot NET Framework minimálně ve verzi 4.0, program se spustí poklepáním na exe soubor.

Po spuštění aplikace je nutné nejdříve inicializovat zařízení, to provedeme kliknutím na „Nastavení“ v horním menu. Vybereme příslušný COM port, popřípadě Baud rate (u zdroje je napevno přednastavená na 19200) a potvrdíme stisknutím tlačítka „Inicializuj port“. Pokud se zdroj nenachází ani na jednom z portů pro výběr, je nutné ho ve „Správci zařízení“ systému Windows, nastavit na jeden z dostupných portů.



Obr. 23: Nastavení COM portu a Baud rate - Obrázek ukazuje nabídku „Nastavení“, ve které lze nastavit COM port na který je zdroj připojen a také Baud rate. Po zatržení příslušného portu a rychlosti nastavení potvrdíme tlačítkem „Inicializuj port“. Zdroj se inicializuje a je připraven k práci.

Pokud je zdroj úspěšně inicializován, můžou být zadávány příkazy a můžeme se zdrojem začít pracovat. Historii příkazu poslaných na zdroj a případné jeho odpovědi, mohou být uloženy do TXT souboru, před ukončením práce. To je dobré například pro zjištění závady. Po zavření aplikace a jejím opětovném spuštění musí být zdroj znovu inicializován.

7.8 Popis zdrojového kódu

Programovací jazyk C# je objektově orientovaný programovací jazyk, a proto je zdrojový kód aplikace rozdělen do jednotlivých tříd. Jelikož MS Visual Studio umožňuje

vytvořit formulářovou aplikaci pro Windows v designeru, část kódu se tedy automaticky generuje. Generují se hlavně vytvořené ovládací prvky a hlavičky jejich metod, samotné metody už programuje programátor. Automaticky vygenerovaný kód se nachází ve třídách „MainForm.Designer.cs“ a „SettingForm.Designer.cs“, kde se nachází především komponenty vytvořené a modifikované pomocí designeru. V těchto třídách však lze dělat změny i ručně.

Ve třídě „Program.cs“ se nachází klasická funkce main, která se spouští při startu aplikace. Funkce main neobsahuje žádné parametry a jsou v ní volány tři funkce pro spuštění formulářové aplikace. První metoda povolí vizuální styly, druhá zajistí, že nové ovládací prvky používají GDI (Graphic Device Interface – slouží k reprezentaci grafických objektů a jejich transformací do výstupního zařízení) podle třídy TextRendered. Metoda run spustí formulářovou aplikaci.

Třída „MainForm.cs“ tvoří jádro aplikace. Je v ní naprogramovaná funkčnost hlavního formulářového okna aplikace. Třída „PortCommunication.cs“ obsahuje vlastní komunikaci počítače se zařízením, přes sériovou linku RS-232. Okno pro nastavení aplikace je naprogramováno v třídě „SettingForm.cs“ k tomu se váže pomocná třída „Setting.cs“.

7.8.1 Třída „MainForm.cs“

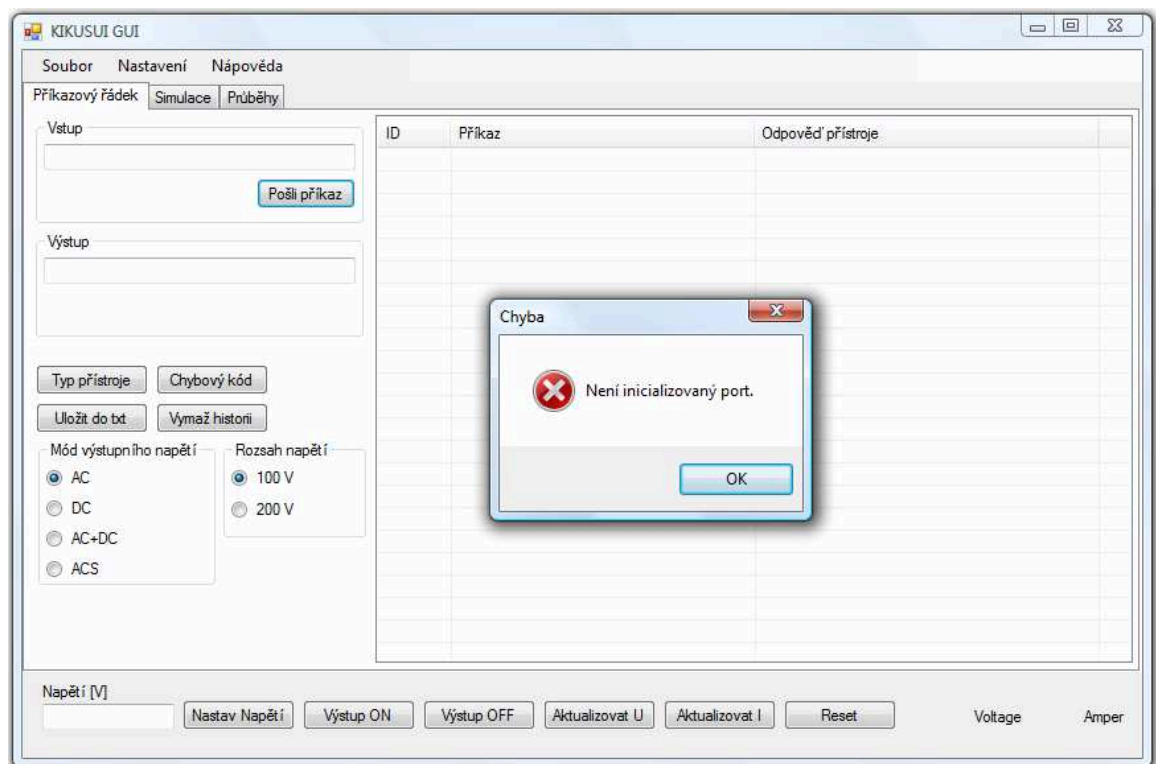
Obsahuje kód hlavního formuláře aplikace, jsou v ní naprogramovány funkce jednotlivých ovládacích prvků aplikace. Kontroluje zadané vstupy (jsou ní odchyceny výjimky).

7.8.2 Třída „PortCommunication.cs“

V této třídě jsem naprogramoval komunikaci se zdrojem přes rozhraní RS-232. Obsahuje metodu Init, která umožňuje inicializaci přístroje. Dále obsahuje metodu SendRequest, pro zaslání příkazu na zařízení, tato metoda vrací odpověď přístroje. Vrácení odpovědi zajišťuje funkce ReadData, která získává odpověď ze zdroje znak po znaku.

8 Testování aplikace

Vytvořenou aplikaci jsem testoval na dvou počítačích a to na stolním PC s operačním systémem Windows XP a nainstalovaným NET Framework ve verzi 4.0 a na notebooku HP ProBook 4710s s operačním systémem Windows 7. Testování probíhalo spuštěním aplikace, inicializací zdroje a odesláním několika příkazů na zdroj. Rovněž jsem se snažil testovat časté chyby při zadávání (např. zkusit se zdrojem komunikovat, když není inicializovaný port) a odchytil je.



Obr. 24: Ukázka odchytení chyby, pokud se budeme snažit se zdrojem komunikovat, aniž by byl inicializovaný port, bude tato chyba odchycena.

8.1 Testování vstupů a výstupů

Test probíhal posíláním příkazů na zdroj a odečítáním odpovědí pomocí funkce ReadData, nacházející se v PortCommunication. Po zaslání takového příkazu, po kterém zdroj vrací nějakou odpověď, se tato odpověď zobrazí v textovém poli „Výstup“. Vše fungovalo a tak po zaslání několika příkazů jsem testování ukončil.

Tab. 3: Tabulka ukazuje příklady vstupů a výstupů

Vstup	Výstup (pokud existuje)	Poznámka
*RST	-	Resetuje všechna nastavení zdroje.
*IDN?	KIKUSUI ELECTRONICS CORP.,PCR1000LA,0,3.10	Vrátí typ přístroje.
VOUT?	29.9	Vrátí aktuálně nastavenou hodnotu napětí ve voltech.

8.2 Testování simulací poruchových stavů

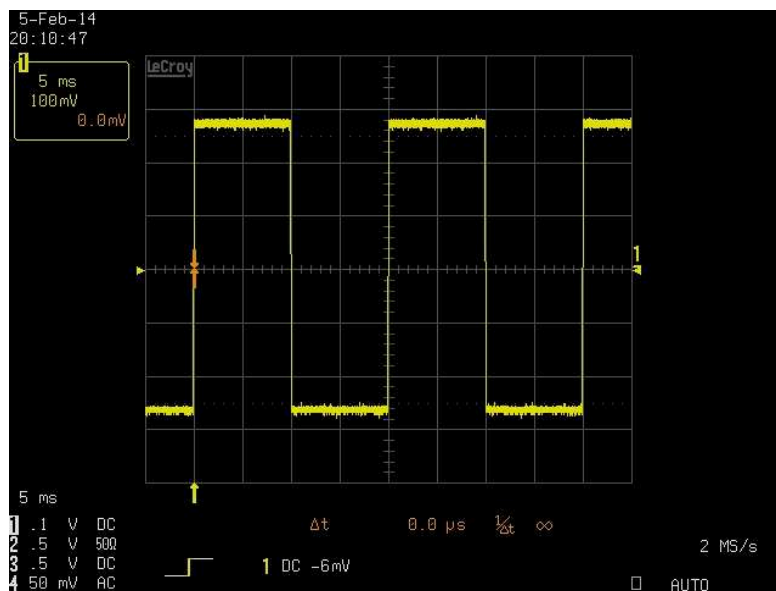
Při testování funkčnosti simulací poruchových stavů jsem provedl opět několik testů. Testy proběhly v pořádku a výsledky jsem zobrazil na osciloskopu.

Tab. 4: Tabulka ukazuje příklad testu poruchového stavu, v tomto případě přepětí napětí a jeho nastavené hodnoty

Veličina	Nastavená hodnota
T1 [deg]	90
T2 [s]	0.1
T3 [s]	0.1
T4 [s]	0.1
T5 [s]	0.1
Opakování	30
Normální napětí [V]	20
Napětí T3[V]	50

8.3 Nahrávání průběhů napájecích napětí do bank zdroje

Pro tento test jsem naprogramoval požadované průběhy napájecích napětí a ty pak nahrál do zdroje. Nahrané průběhy jsem následně zobrazil na osciloskopu. Ukázka průběhu viz obrázek 25, zbytek viz kapitola 5.



Obr. 25: Ukázka průběhu – obdélníkový signál

9 Závěr

V závěru budu hodnotit výsledky bakalářské práce „Generátor libovolných průběhů napájecích napětí.“ Práci jsem pojal jak z teoretického, tak z praktického hlediska, práci jsem si tedy rozdělil na dvě části a to část teoretickou a část praktickou. V teoretické části jsem se zabýval spíše elektrotechnickými a hardwarovými věcmi a v části praktické jsem se zaměřil hlavně na samotnou implementaci mé aplikace pro ovládání zdroje.

9.1 Zhodnocení mé práce

Na začátku mé práce jsem se nejprve musel seznámit se zdrojem Kikusui PCR 1000 LA a s jeho funkcemi. Dále jsem musel načerpat taky teoretické znalosti ze základů virtuální instrumentace, teorii průběhů napájecích napětí a s tím souvisejících poruchových stavů a něco málo z teorie hardwaru. Pro načerpání znalostí o sériovém rozhraní RS 232 a vůbec o sériových rozhraních obecně, jsem použil velice dobrou a pro mé účely dostačující knihu o hardwaru[4], kde je teoreticky vše popsáno.

Po nastudování veškeré teorie, jsem začal dělat první pokusy a měření se zdrojem. První má zkušenost s ovládáním zdroje přes počítač byla v programu LabWindows/CVI, kde jsem se naučil základní příkazy a funkce zdroje, také jsem si zkusil do zdroje poslat testovací průběhy napětí.

Když jsem ukončil testování v programu LabWindows/CVI, hledal jsem vhodné programové prostředí a vhodný jazyk pro implementaci svého GUI. Nakonec jsem se rozhodl pro MS Visual Studio ve verzi 2010 (v té době byla sice již verze 2012, ale z důvodu dobrých zkušeností s verzi 2010 jsem se rozhodl právě pro ni) a programovací jazyk C#, který je pro GUI aplikace ideální.

9.2 Dosažené výsledky

Výstupem mé práce je grafické uživatelské rozhraní pro ovládání zdroje s počítače, print screeny z osciloskopu a tento text. V práci jsem dosáhl toho, že jsem vytvořil uživatelské rozhraní a s jeho pomocí jsem do jednotlivých bank zdroje poslal naprogramované průběhy. Tyto průběhy jsem pak zobrazil na osciloskopu a následně uložil do počítače a dále zpracoval. Dále jsem naprogramoval a provedl simulace poruchových stavů a to přepětí, podpětí a přerušení napájení. Poruchové stavy jsem rovněž zaznamenal pomocí osciloskopu a přenesl do počítače.

Přínos v mé práci bych viděl v tom, že dosud neexistoval žádný program pro takovéto ovládání zdroje, který by byl volně dostupný. Rovněž při poruchových stavech je možno zdroj ovládat jedinečně přes počítač a proto je dobré tuto aplikaci mít. Pomocí mé vytvořené aplikace může být zdroj ovládán na dálku pomocí příkazů zadávaných v aplikaci. Aplikace může být rovněž využita při výuce a to v předmětu Elektromagnetická kompatibilita.

9.3 Shrnutí

- Seznámil jsem se s programovatelným zdrojem Kikusui PCR 1000 LA a naučil jsem se ho ovládat přes RS-232.
- Implementoval jsem uživatelské rozhraní pro ovládání zdroje a generoval jsem zadané průběhy napětí.
- Provedl jsem porovnání napájení ze sítě a napájení z UPS. Zapojil jsem PC na zdroj a změřené hodnoty jsem porovnal s harmonickým napájením.

Literatura

- [1] KAINKA, Burkhard. *Elektronika s podporou PC: Visual Basic v praxi*. 1. české vyd. Ostrava: HEL, 2004. ISBN 80-861-6722-4.
- [2] *Manuál zdroje PCR 1000 LA*. Dostupné z:
http://www.kikusui.co.jp/kiku_manuals/P/PCR_LA_EB.PDF
- [3] ŽÍDEK, Jan. Virtuální instrumentace.
- [4] HORÁK, Jaroslav. *Hardware: učebnice pro pokročilé*. 4. aktualiz. vyd. Brno: Computer Press, 2007. ISBN 978-80-251-1741-5.
- [5] FAJKUS, Marcel. *Modulace signálů a jejich vliv na spektrum signálu* [online]. [cit. 2014-03-13]. Dostupné z: <http://modulace.imatte.cz/teorie.php>
- [6] PIRKL, Josef. *Řešené příklady v C sharp: aneb C# skutečně prakticky*. KOPP.
- [7] VIRIUS, Miroslav. *Od C k C#*. 1. vyd. České Budějovice: Kopp, 2002. ISBN 80-723-2176-5.
- [8] VIRIUS, Miroslav. *Jazyky C a C: kompletní průvodce*. 2., aktualiz. vyd. Praha: Grada, 2011, 367 s. ISBN 978-80-247-3917-5.
- [9] Lukeš, P.: *Poklesy napětí v síti nn a spolehlivost výpočetní a řídicí techniky*, AUTOMA 1/2008.
- [10] ČSN 33 0120. *Elektrotechnické předpisy - Normalizovaná napětí IEC*.
- [11] ČSN EN 50160. *Charakteristiky napětí elektrické energie dodávané z veřejných distribučních sítí*.
- [12] ČSN EN 61 000-4-7. *Elektromagnetická kompatibilita (EMC) - Část 4-7: Zkušební a měřicí technika - Všeobecná směrnice o měření a měřicích přístrojích harmonických a meziharmonických pro rozvodné sítě a zařízení připojovaná do nich*.

Seznam příloh

- I. Manuál vytvořené aplikace
- II. Obrázek z aplikace – záložka „Simulace“
- III. Obrázek z aplikace – záložka „Průběhy“
- IV. Třídní diagram aplikace
- V. Zdrojový kód aplikace